

# An Energy-Efficient Data Collection Scheme Using Denoising Autoencoder in Wireless Sensor Networks

Guorui Li, Sancheng Peng\*, Cong Wang, Jianwei Niu, and Ying Yuan

**Abstract:** As one of the key operations in Wireless Sensor Networks (WSNs), the energy-efficient data collection schemes have been actively explored in the literature. However, the transform basis for sparsifying the sensed data is usually chosen empirically, and the transformed results are not always the sparsest. In this paper, we propose a Data Collection scheme based on Denoising Autoencoder (DCDA) to solve the above problem. In the data training phase, a Denoising AutoEncoder (DAE) is trained to compute the data measurement matrix and the data reconstruction matrix using the historical sensed data. Then, in the data collection phase, the sensed data of whole network are collected along a data collection tree. The data measurement matrix is utilized to compress the sensed data in each sensor node, and the data reconstruction matrix is utilized to reconstruct the original data in the sink. Finally, the data communication performance and data reconstruction performance of the proposed scheme are evaluated and compared with those of existing schemes using real-world sensed data. The experimental results show that compared to its counterparts, the proposed scheme results in a higher data compression rate, lower energy consumption, more accurate data reconstruction, and faster data reconstruction speed.

**Key words:** wireless sensor networks; data collection; neural networks; autoencoder; data reconstruction

## 1 Introduction

As one of the core components of cyber physical systems, Wireless Sensor Networks (WSNs) have gained much attention from both industrial and research communities. A typical WSN is usually composed of hundreds or even thousands of tiny, inexpensive sensor nodes that can sense the surroundings and transmit the sensed data back to the sink node. The representative

applications of WSNs include environment monitoring, intelligent transportation, industrial and agricultural automation, smart home, and military surveillance<sup>[1]</sup>.

Data collection is one of the core functions in WSNs. All sensor nodes periodically perform sensing and transmitting operations in a distributed and cooperative manner to acquire the sensed information of the surveillant area<sup>[2]</sup>. The sensed data collected from the network exponentially grow with time. The exponentially grown data exhibits prominent characteristics of big data, such as high volume, high velocity, and high variety. According to a report by Oracle, the volume of data generated by WSNs is expected to be in the order of petabytes. However, the sensed big data are not efficiently managed, especially in the data collection process<sup>[3]</sup>. Furthermore, wireless sensor nodes are usually powered by batteries, which are drained quickly and are hard or even impossible to be replaced. The limited computational and communication capacities of wireless sensor nodes

• Guorui Li, Cong Wang, and Ying Yuan are with School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China. E-mail: lgr@neuq.edu.cn; congw@neuq.edu.cn; yuanying@neuq.edu.cn.

• Sancheng Peng is with School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China. E-mail: psc346@aliyun.com.

• Jianwei Niu is with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China. E-mail: niujianwei@buaa.edu.cn.

\* To whom correspondence should be addressed.

Manuscript received: 2017-05-10; revised: 2017-07-14; accepted: 2017-08-07

further aggravate the difficulty of data collection. Therefore, the design of energy-efficient data collection schemes for big data still remains one of the key research issues in WSNs.

Currently, several data collection schemes for WSNs have been proposed in Ref. [4]. Based on the underlying techniques adopted in the schemes, state-of-the-art data collection schemes can be classified into four categories: signal processing-based scheme, routing-based scheme, information theory-based scheme, and compressed sensing-based scheme. Of these data collection schemes, the compressed sensing-based scheme has attracted the most attention from researchers, and it has gradually become the research focus of the data collection schemes for WSNs.

Compressed Sensing (CS), also known as compressive sampling, is a new suite of signal processing theories and techniques that was introduced by Donoho<sup>[5]</sup> in 2006 and Candes and Wakin<sup>[6]</sup> in 2008, respectively. It is based on the principle that a sparse signal can be reconstructed from far fewer samples than those required by the classical Shannon-Nyquist sampling theorem by finding the sparsest solution to the underdetermined linear systems. The two essential conditions under the signal reconstruction are sparsity and incoherence<sup>[7]</sup>. The compressibility and computational asymmetry of the CS theory have made data collection of resource-constrained WSNs very practicable.

In recent years, several CS-based data collection schemes have been proposed to minimize the volume of transmitted data and prolong the lifetime of the whole sensor network. Luo et al.<sup>[8]</sup> proposed the first practical design of CS-based data collection scheme by compressing the sensed data of each sensor node and adding them along the data transmission routes toward the sink. In Ref. [9], Luo et al. advocated that the performance of the naively designed CS-based data collection scheme is worse than that of non-CS-based data collection scheme; thus, they proposed a data collection scheme based on hybrid CS by combining the traditional non-CS-based data collection techniques with the CS theory. In Ref. [10], an adaptive data collection scheme was introduced by integrally fusing the autoregressive model with the CS theory integrally. The inherent spatio-temporal correlation among the sensed data of a whole sensor network leads to the matrix completion technique, which is a natural extension of the CS theory; this has also

been utilized to collect the WSNs sensed data. For instance, Cheng et al.<sup>[11]</sup> proposed a spatio-temporal compressed data gathering scheme based on the spatial and temporal similarity and short-term stability of the sensed data. Kong et al.<sup>[12]</sup> proposed a sensed data matrix completion method by exploring the low rankness and multiple attributes correlation of the sensed data in WSNs. Xiang et al.<sup>[13]</sup> proposed a dual-level data reconstruction algorithm by combining the matrix completion with a fine-tuned CS technique.

However, the researchers in the CS community usually ignored an important problem. It is well known that most natural signals are not sparse in their original forms; therefore, they should not be sparsified through a predefined basis transformation. The usually chosen transform bases include Discrete Cosine Transform (DCT) basis, discrete fourier transform basis, and discrete wavelet transform basis<sup>[14]</sup>. The chosen process of the transform basis is usually empirical, and the transformed signal under the selected transform basis is not always the sparsest.

To obtain the sparsest transformed signal and consequently reduce the amount of transmitted data within the network, we designed a Data Collection scheme based on Denoising Autoencoder (DCDA) in this study. The proposed DCDA scheme includes two correlated phases: the data training phase and the data collection phase. In the data training phase, a data measurement matrix and a data reconstruction matrix are obtained by training the Denoising AutoEncoder (DAE) using the historical surveillant data. The obtained data measurement matrix can be regarded as the trained sparse transform basis, which is suitable for the subsequent sensed data. In the data collection phase, a data collection tree is built, and all sensed data are collected in a hybrid and cooperative manner. By comparing the DCDA scheme with the well-known DCT basis and other state-of-the-art data collection schemes based on numerical experiments, we detailedly evaluated and analyzed the performance of the proposed scheme in this study.

The rest of this paper is organized as follows. In Section 2, we introduce the basic concepts of Artificial Neural Networks (ANNs) and autoencoders, and in Section 3, we describe the proposed scheme. In Section 4, we demonstrate the experimental results which include the data compression performance and the data reconstruction performance. In Section 5, we state our conclusions.

## 2 Artificial Neural Network and Autoencoder

Artificial neural networks, also known as connectionist systems, are key tools that have been widely used in machine learning. They can extract meaningful features and solve certain problems that are usually difficult or even impossible to explicitly program. The prevalent deep learning is also built upon ANNs with deep architectures. Typical applications of ANNs include computer vision, speech recognition, machine translation, autonomous driving, and recommendation systems.

An ANN is composed of many neurons that are arranged in different layers. The first and last layers are called input and output layers, respectively, and the intermediate layers are called hidden layers. Each neuron takes an  $(n + 1)$ -dimensional vector  $\mathbf{x} = \{x_0, x_1, \dots, x_n\}$  as its input and multiplies it with an  $(n + 1)$ -dimensional weight vector  $\mathbf{w} = \{w_0, w_1, \dots, w_n\}$  to compute the pre-activation result  $\mathbf{w}^T \mathbf{x}$ , where  $x_0$  is fixed to 1, and  $w_0$  is usually called bias. Then, the pre-activation result  $\mathbf{w}^T \mathbf{x}$  is fed into a nonlinear activation function  $f: \mathbf{R} \rightarrow \mathbf{R}$  to compute the neuron activation  $f(\mathbf{w}^T \mathbf{x})$ . The available nonlinear activation functions include sigmoid function, tanh function, and rectified linear units function.

The commonly designed ANNs are usually organized in several layers, where the outputs of the neurons in one layer are fed into the neurons in the next layer as inputs. A three-layer ANN is shown in Fig. 1. For convenience, we describe the network parameters of an  $s$ -layer ANN as a series of weight matrixes  $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \dots, \mathbf{W}^{(s)}\}$ , where  $W_{ij}^{(l)}$  is the

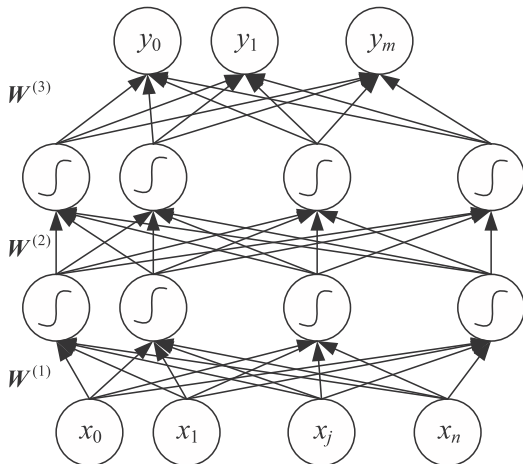


Fig. 1 A three-layer artificial neural network.

weight associated with the connection between neuron  $j$  in layer  $l$  and neuron  $i$  in layer  $l + 1$ . Thus, the activation of neuron  $i$  in layer  $l$ , i.e.,  $a_i^{(l)}$ , can be represented as

$$a_i^{(l)} = f \left( \sum_{k=0}^{n^{(l-1)}} w_{ik}^{(l-1)} a_k^{(l-1)} \right) \quad (1)$$

where  $n^{(l-1)}$  is the number of neurons in layer  $l - 1$ , and the layer index  $l = 2, \dots, s$ . Furthermore, the entire output of ANN in layer  $l$ , i.e.,  $\mathbf{a}^{(l)}$ , can be represented as

$$\mathbf{a}^{(l)} = f \left( \mathbf{W}^{(l-1)} \mathbf{a}^{(l-1)} \right) \quad (2)$$

The lowest output  $\mathbf{a}^{(1)}$  is actually the input vector  $\mathbf{x}$  of the network.

To train an ANN, a forward propagation process is first executed by computing the output of an ANN from the bottom-up until the output layer is reached. Then, the output of the network is compared with the desired output by a predefined loss function, and the error is gradually propagated back to the input layer. Meanwhile, the corresponding weights are also updated by the well-known back propagation method<sup>[15]</sup>. After a certain number of iterations, the trained ANN can approximate the mapping relationship between the input and output layers with a high accuracy<sup>[16]</sup>.

Autoencoders are an unsupervised learning framework in ANNs<sup>[17]</sup>. Their main function is to learn a low-dimensional representation of the original high-dimensional data and compress them through dimensionality reduction. First, an autoencoder maps the input data  $\mathbf{x} \in [0, 1]^{n+1}$  to a hidden representation  $\mathbf{y} \in [0, 1]^{m+1}$  with an encoder, that is,

$$\mathbf{y} = f(\mathbf{W}\mathbf{x}) \quad (3)$$

where  $m$  is less than  $n$  in most cases. Then, the latent representation  $\mathbf{y}$  is mapped back to a reconstructed data  $\hat{\mathbf{x}} \in [0, 1]^{n+1}$  with a decoder, that is,

$$\hat{\mathbf{x}} = f(\mathbf{W}'\mathbf{y}) \quad (4)$$

Therefore, the network parameters of an autoencoder include two weight matrixes:  $\{\mathbf{W}, \mathbf{W}'\}$ . When the reverse mapping weight matrix  $\mathbf{W}' = \mathbf{W}^T$ , the weight matrixes of the autoencoder are usually referred as tied weights.

In an autoencoder, the difference between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  can be utilized to build the loss function. By minimizing the loss function with training iterations, the hidden representation  $\mathbf{y}$  can be regarded as a compressed form of original data  $\mathbf{x}$  that captures its inherent main features. It has been proved that an autoencoder has the

same feature extraction performance as the principal component analysis method when the activation function  $f$  is linear. Moreover, if the activation function  $f$  is nonlinear, an autoencoder can capture the multi-modal aspects of the input data<sup>[18]</sup>.

In recent years, several variations of autoencoders have been proposed in the literature. The representative variations include DAE<sup>[19]</sup>, sparse autoencoder, contractive autoencoder<sup>[20]</sup>, and variational autoencoder<sup>[21]</sup>. Among the proposed autoencoder variations, DAE has been proved most capable of learning robust features from the input data. Moreover, it is also easily implemented by simply injecting some random corruptions to the training data. We utilize DAE in our proposed data collection scheme, and in the next section we describe it in detail.

### 3 Data Collection Scheme Based on Denoising Autoencoder

The DCDA scheme includes two correlated phases: the data training phase and the data collection phase. In the data training phase, we train the DAE with the historical sensed data. Then, the weight matrixes in the encoder and decoder can be utilized to compress the subsequent sensed data and reconstruct the original sensed data, respectively. In the data collection phase, each sensor node compresses the sensed data and forward the compressed data to the sink along the data collection tree in a hybrid and cooperative manner. Then, the original sensed data are reconstructed in the sink using the decoder part of the DAE.

#### 3.1 Data training phase

In the data training phase, we train the DAE with the historical surveillant data in an off-line mode. The core idea behind the DAE is to discover more robust features of the input data from its corrupted version. In WSNs, the collected sensed data are usually influenced by all sorts of noise or even a poor wireless connectivity. Hence, we try to recover the uncorrupted sensed data or the missing data from the corrupted data which were collected in the sink.

Essentially, a DAE can be seen as a stochastic variation of an autoencoder. An instance of DAE is shown in Fig. 2. First, the input data  $\mathbf{x}$  is normalized and then converted to a corrupted version  $\tilde{\mathbf{x}}$  by randomly masking some entries of  $\mathbf{x}$  to 0. The degree of corruption is controlled by the corruption level  $\nu$ . The larger the  $\nu$  is, the more sensed data are set to 0.

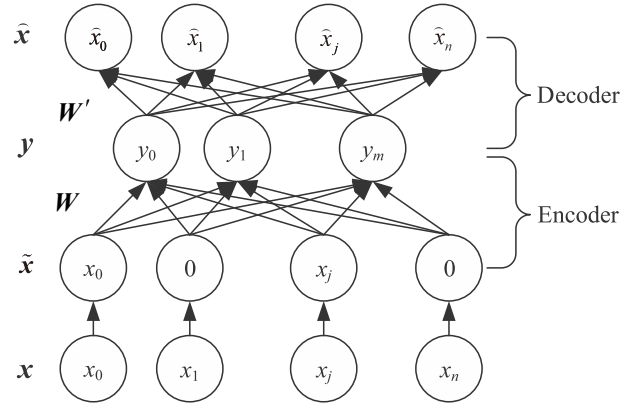


Fig. 2 An instance of the denoising autoencoder.

After that, the encoder part of the DAE compresses the corrupted data  $\tilde{\mathbf{x}}$  to a lower-dimensional data  $\mathbf{y}$  with the weight matrix  $\mathbf{W}$ ; that is,  $\mathbf{y} = f(\mathbf{W}\tilde{\mathbf{x}})$ . Then, the decoder part of the DAE reconstructs the original high-dimensional data  $\hat{\mathbf{x}}$  from the compressed data  $\mathbf{y}$  with the weight matrix  $\mathbf{W}'$ , that is,  $\hat{\mathbf{x}} = f(\mathbf{W}'\mathbf{y})$ .

In the proposed DCDA scheme, we use the well-known sigmoid function as the nonlinear activation function.

$$f(z) = \frac{1}{1 + e^{-z}} \quad (5)$$

Meanwhile, the squared error loss function is utilized to measure the data reconstruction performance of the DAE.

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (6)$$

By repeatedly executing the mini-batch stochastic gradient descent algorithm<sup>[22]</sup>, the difference between the original sensed data  $\mathbf{x}$  and the reconstructed sensed data  $\hat{\mathbf{x}}$  decreases gradually. Meanwhile, the corresponding network parameter  $\{\mathbf{W}, \mathbf{W}'\}$  of the DAE is also adjusted accordingly.

The trained DAE is separately utilized to compress and then reconstruct the WSNs sensed data. Specifically, the weight matrix  $\mathbf{W}$  can be regarded as the data measurement matrix and utilized to compress the sensed data in each sensor node. After collecting all the compressed data, the weight matrix  $\mathbf{W}'$  can be regarded as the data reconstruction matrix and utilized to reconstruct the original sensed data in the sink node. In the next subsection, we describe the data collection phase in detail.

#### 3.2 Data collection phase

In the data collection phase, a data collection tree should first be built. Currently, several data collection tree-building algorithms have been proposed in the

literature. In this study, we collected the sensed data by using the group-based data collection tree which we proposed in Ref. [23]. Specifically, we can divide the whole sensor network into several data correlated groups by the refined  $\delta$ -grouping algorithm. Then, each sensor node  $i$  transmits its sensed data  $x_i$  to its parent  $p_i$  which is appointed in the grouping process. Thus, the sensed data of the whole sensor network can be collected progressively in the sink node by following the parent indicator of each sensor node.

In non-compression data collection schemes, each sensor node transmits its sensed data as well as the sense data received from its children to its parent in the data collection tree. Although the whole data collection operations are easily implemented, the energy and bandwidth consumptions of the sensor nodes that are closer to the sink are remarkably higher than those of other sensor nodes. In other words, the sensor nodes adjacent to the sink usually run out of energy faster than other sensor nodes.

To reduce the amount of transmitted data and avoid the aforementioned energy depletion problem, we designed a data collection algorithm based on DAE. Each sensor node runs the algorithm distributedly and compresses the sensed data in a hybrid and cooperative mode. Its pseudo code is described in Algorithm 1.

In the above algorithm, each sensor node sends a tuple (label, data) to its parent. If the label is set to 0, the sensed data are transmitted in an uncompressed form; otherwise, the sensed data are transmitted in a compressed form. The data collection operation of the leaf node is rather simple. Its only function is to transmit the uncompressed sensed data  $x_i$  as well as the corresponding label 0 to its parent  $p_i$  (line 2). For the non-leaf node, there are two different situations. After

receiving the sensed data  $\{(l_{c_i^1}, x_{c_i^1}), \dots, (l_{c_i^k}, x_{c_i^k})\}$  from its children  $\{c_i^1, \dots, c_i^k\}$  (line 4), sensor node  $i$  sends the combined sensed data to its parent  $p_i$ . When the number of children  $k$  is less than  $m - 1$ , sensor node  $i$  only needs to send its sensed data  $x_i$  combined with those of its children  $\{x_{c_i^1}, \dots, x_{c_i^k}\}$  to  $p_i$  in an uncompressed form (line 6); otherwise, sensor node  $i$  computes the compressed data  $x_i W_i + \sum_{j=1}^k \left( (1 - l_{c_i^j}) x_{c_i^j} W_{c_i^j} + l_{c_i^j} x_{c_i^j} \right)$  and sends it to  $p_i$  (line 8), where  $W_i$  is the  $i$ -th column of weight matrix  $W$ . The above compressed data are composed of three parts. The first part  $x_i W_i$  is the compressed sensed data  $x_i$  of node  $i$ . The second part  $\sum_{j=1}^k (1 - l_{c_i^j}) x_{c_i^j} W_{c_i^j}$  is the compressed sensed data sent from node  $i$  children. The last part  $\sum_{j=1}^k l_{c_i^j} x_{c_i^j}$  is the already compressed sensed data sent to node  $i$ .

An instance of the DCDA algorithm is shown in Fig. 3. The number of nodes in the hidden layer  $m$  is set to 4. Therefore, non-leaf nodes 3 and 15 send their sensed data as well as those received from their children to their parents in an uncompressed form. On the contrary, non-leaf nodes 7, 12, 8, and 16 send their sensed data and those of their children to their parents in a compressed form. Finally, the sink receives all sensed data of the whole sensor network in a compressed form, i.e.,  $\sum_{j=1}^{16} x_j W_j$ .

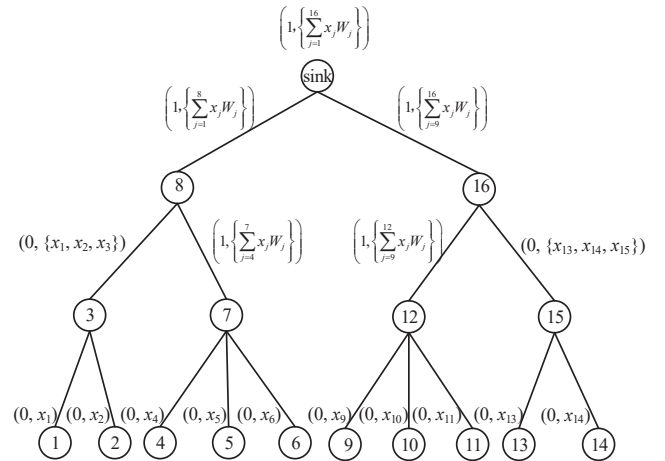
Only few vector additions and scalar multiplications of vector are required in the non-leaf nodes, and there is neither non-linear activation nor data decompression in the resource-constrained wireless sensor nodes. Meanwhile, the number of transmitted data in each sensor node is less than or equal to  $m$ , which is the

---

**Algorithm 1** Data collection algorithm based on denoising autoencoder

---

- 1: **if**  $i$  is a leaf node **then**
  - 2:   send  $(0, x_i)$  to  $p_i$
  - 3: **else**
  - 4:   receive  $\{(l_{c_i^1}, x_{c_i^1}), \dots, (l_{c_i^k}, x_{c_i^k})\}$  from its children
  - 5:   **if**  $k < m - 1$  **then**
  - 6:     send  $(0, \{x_i, x_{c_i^1}, \dots, x_{c_i^k}\})$  to  $p_i$
  - 7:   **else**
  - 8:     send  $\left( 1, x_i W_i + \sum_{j=1}^k \left( (1 - l_{c_i^j}) x_{c_i^j} W_{c_i^j} + l_{c_i^j} x_{c_i^j} \right) \right)$   
to  $p_i$
  - 9:   **end if**
  - 10: **end if**
- 



**Fig. 3** An instance of the data collection algorithm based on denoising autoencoder.

number of nodes in the DAE hidden layer. Therefore, the energy consumption for each sensor node is quite low.

When the sink node receives all the compressed data from its directly connected children, it can compute the compressed data  $z$  of the whole sensor network by adding them together, that is,

$$z = \sum_{i=1}^n x_i W_i \quad (7)$$

It is the same as the pre-activation result  $\mathbf{W}^T \mathbf{x}$  in the encoder part of the DAE. Thus, we can easily compute the reconstructed sensed data  $\hat{\mathbf{x}}$  by inputting  $z$  into the activation function  $f$  and then decoding it with the decoder part of the DAE. In other words, the reconstructed sensed data of the sensor network can be simply computed as follows:

$$\hat{\mathbf{x}} = f(\mathbf{W}' f(z)) \quad (8)$$

In the CS theory, the data reconstruction process involves several iterations in all proposed data reconstruction algorithms. For example, the support set of the original signal can only be found by successive iterations in the Orthogonal Matching Pursuit (OMP) algorithm<sup>[24]</sup>. However, in the data reconstruction process of our proposed DCDA scheme, many iterations are not needed, and only one matrix-vector multiplication and two non-linear activation functions are required for computing. Therefore, the computational complexity and data reconstruction speed of our proposed scheme are superior to those of CS-based data collection schemes.

## 4 Experiments and Analysis

To evaluate the performance of our proposed scheme, we carried out several numerical experiments based on a real-world sensed dataset. The experimental results are reported and analyzed as follows.

### 4.1 Experiment settings

We used the Intel Berkeley Research Lab WSN dataset<sup>[25]</sup> in the experiments. The temperature, humidity, and light intensity of 54 Mica2Dot sensor nodes were collected between February 28, 2004 and April 5, 2004 on a 30-second cycle. We excluded five sensor nodes from the experiments because of malfunction or energy depletion.

We used the following energy consumption model, which was also adopted in Ref. [26] and other related researches, to evaluate the consumed energy in wireless sensor nodes.

$$E_T(k, d) = \begin{cases} k(E_{Tx} + d^2 E_{Amp}), & \text{if } d < d_T; \\ k(E_{Tx} + d^4 E_{Amp}), & \text{if } d \geq d_T \end{cases} \quad (9)$$

$$E_R(k) = k E_{Rx} \quad (10)$$

In Eq. (9),  $E_T(k, d)$  represents the energy consumed in transmitting  $k$  (bits) data to a receiver  $d$  (meters) away. If the distance between transmitter and receiver is less than the predefined distance threshold  $d_T$ , the free space power loss channel mode is utilized; otherwise, the multiple path fading power loss channel model is utilized. In the experiments, the distance threshold  $d_T$  was set to 75 m. Moreover,  $E_{Tx}$  and  $E_{Amp}$  represent the energy consumed by the transmitting circuit and the power amplifying circuit, respectively, to process 1 bit data in the data transmission process. Their values were set to 100 nJ/bit and 0.01 nJ/(bit·m<sup>2</sup>), respectively. In Eq. (10),  $E_R(k)$  represents the energy consumed to receive  $k$  (bits) data from the transmitter. The energy consumed to receive 1 bit data from the transmitter is  $E_{Rx}$ , and its value was set to 120 nJ/bit.

In the data training phase, the Theano framework 0.9.0 was utilized to train the DAEs. The elements in the encoder matrix  $\mathbf{W}$  and the decoder matrix  $\mathbf{W}'$  were different. Their initial values were set as random numbers drawn from  $U[-b, b]$  where  $b = \sqrt{6/(H_k + H_{k-1})}$  and  $H_k$  is the number of neurons in the  $k$ -th layer. In the data collection phase, the TOS\_Msg structure of the TinyOS was adopted to encapsulate the sensed data of each sensor node. Specifically, extra 7 bytes of data (address: 2 bytes, type: 1 byte, group: 1 byte, length: 1 byte, cyclic redundancy check: 2 bytes) are required to transmit in addition to the sensed data which occupies 2 bytes.

### 4.2 Data training experiments

In the data training experiments, we selected a subset of original temperature data at an interval of 5 min to reduce the total number of sensed data. Then, the chosen sensed dataset was divided into the training dataset, the validation dataset, and the testing dataset with proportions of 50%, 25%, and 25%, respectively. A subset of the original temperature data is demonstrated in Fig. 4.

In DAE, the number of neurons in the hidden layer  $m$  is a very important parameter. It reflects the dimensions of the key features in the input data. Obviously, the smaller the  $m$ , the larger the compression rate DAE can provide. Thus, it is equivalent to the sparsity of

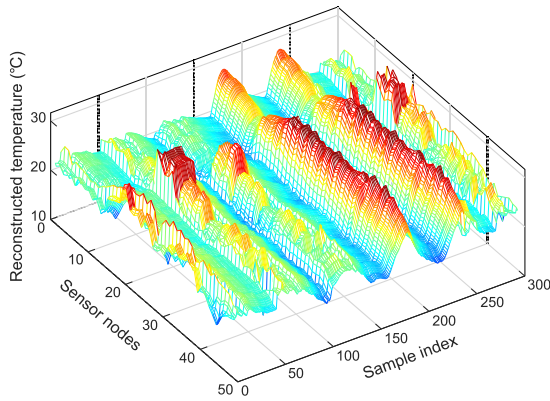


Fig. 4 A subset of the original temperature data.

the signal in the CS theory. The average reconstructed temperature errors at different numbers of hidden nodes and sparsity are presented in Fig. 5. The DCT was utilized to sparsify the sensed data, and the sparsity was obtained by discarding some small coefficients of the transformed result. Then, we reconstructed the temperature data by applying the inverse DCT on the truncated signal. The average data reconstruction error of the DAE was less than that of DCT when the number of hidden nodes or sparsity was above 3 (Fig. 5); meanwhile, the average reconstructed temperature error was less than 0.6°C under the same condition. In other words, compared to the DCT, more sparse features can be extracted by using the DAE with the same acceptable data reconstruction accuracy.

To measure the data reconstruction performance of the DAE without considering the scale of the input data, we utilized the following Signal-to-Noise Ratio (SNR) measurement.

$$SNR = 10 \log_{10} \frac{\|x\|_2^2}{\|x - \hat{x}\|_2^2} \quad (11)$$

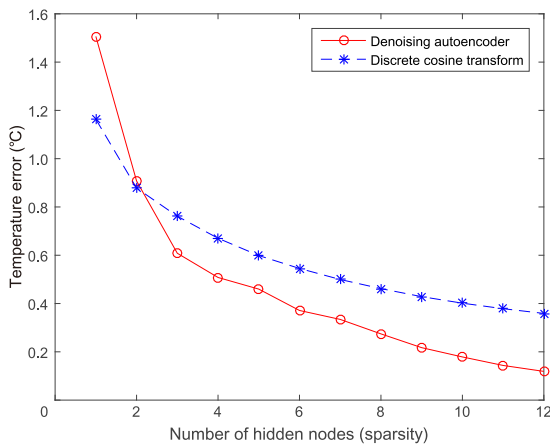


Fig. 5 Average reconstruction error of the denoising autoencoder and discrete cosine transform.

The SNRs of the DAE at different numbers of hidden nodes and training times (Fig. 6) show that the SNR of the DAE increased with the data training times. Thus, more accurate data reconstruction result can be obtained by increasing the data training times. Furthermore, the SNR of the DAE increased with the number of hidden nodes and then gradually stabilized. Therefore, we can reduce the number of hidden nodes to increase the data compression rate of the DAE.

### 4.3 Data collection experiments

The numbers of transmitted data of the proposed DCDA scheme, Compressive Data Gathering (CDG) scheme<sup>[8]</sup>, and non-compression scheme for one round of data collection operation are shown in Fig. 7. The corresponding average energy consumption of sensor node is also shown in Fig. 8. First, we can see that the number of transmitted data and the average energy consumption of each sensor node increased with the number of hidden nodes in the DCDA scheme or with the sparsity in the CDG scheme. The reason behind this

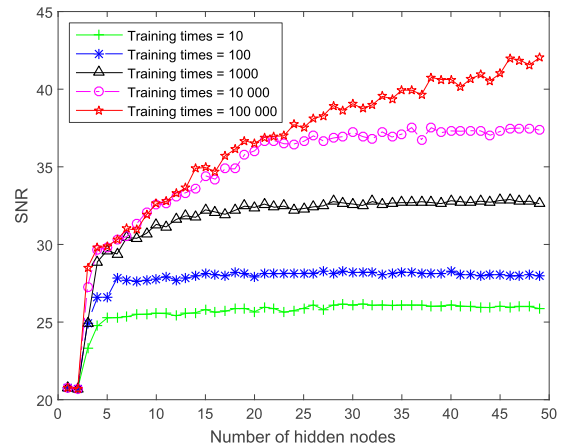


Fig. 6 SNR of the denoising autoencoder.

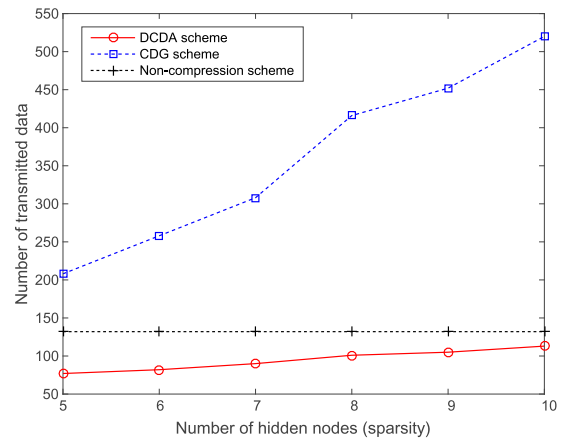
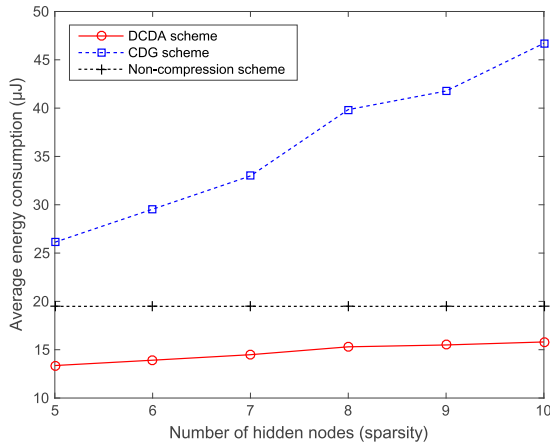


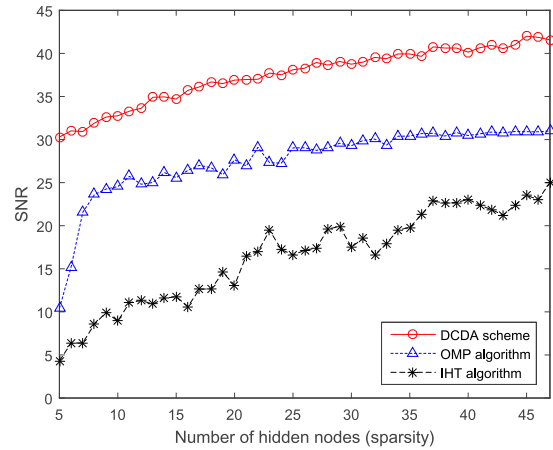
Fig. 7 Number of transmitted data in one round of data collection.



**Fig. 8** Average energy consumption of sensor node in one round of data collection.

phenomenon is clear. The larger the number of hidden nodes or sparsity, the more data are transmitted within the network, and then each sensor node consumes more energy. On the contrary, the number of transmitted data and the average energy consumption are invariant in the non-compression data collection scheme because the sensed data are not compressed in the data collection process. Furthermore, the number of transmitted data and the energy consumption of the DCDA scheme are the lowest among the three data collection schemes. In other words, the data compression performance and the energy efficiency of the proposed DCDA scheme are superior to those of the compressive sensing-based scheme CDG and non-compression data collection scheme.

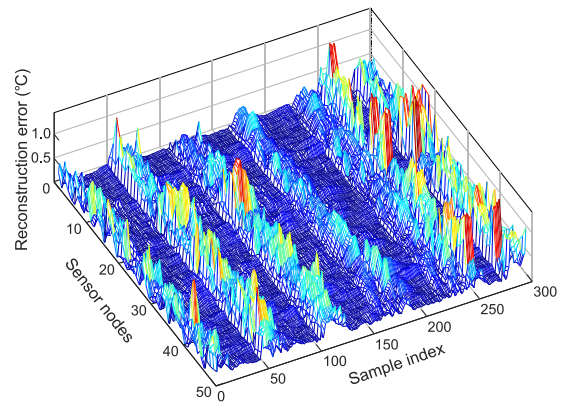
In the proposed DCDA scheme, the sensed data can be reconstructed by the decoder part of the DAE. To measure the data reconstruction performance, we compared it with the OMP algorithm<sup>[24]</sup> and the Iterative Hard Thresholding (IHT) algorithm<sup>[27]</sup>. The OMP and IHT algorithms are data reconstruction algorithms in the CS theory, and they should be combined with the CDG scheme to build a complete CS-based data collection scheme. The SNRs of the DCDA scheme, OMP algorithm, and IHT algorithm (Fig. 9) show that the data reconstruction accuracy of our proposed DCDA scheme was higher than those of the other two algorithms. Meanwhile, the data reconstruction performance of these three schemes (or algorithms) improved with the number of hidden nodes or sparsity. In other words, the smaller the data compression rate, the better the data reconstruction performance these three schemes (or algorithms) can



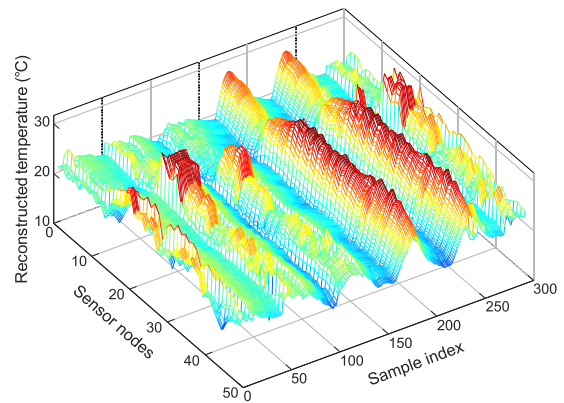
**Fig. 9** SNR of the DCDA scheme, OMP algorithm, and IHT algorithm.

provide.

To further clarify the data reconstruction performance of the DCDA scheme, we show the data reconstruction error of the temperature data in Fig. 10 and the reconstructed temperature data in Fig. 11. When training the DAE, the numbers of hidden nodes and data training operations were 5 and 10 000, respectively. The



**Fig. 10** Reconstruction error of the temperature data.



**Fig. 11** Reconstructed temperature data.



temperature data reconstructed by the DAE was close to the original sensed data (Fig. 11). Meanwhile, most of the reconstruction errors were equal to or near 0°C, and the maximum reconstruction error was less than 1°C. Therefore, the data reconstruction performance of the proposed DCDA scheme is acceptable in most WSN applications. Meanwhile, some reconstruction errors were generated by the DAE, which are visible as the raised peaks in Fig. 10. When the sensed data changes intensively, the DAE cannot precisely represent the sensed data, and then the data reconstruction error would increase.

Furthermore, the data reconstruction time of the DCDA scheme for one round of temperature data was 0.0006 s. It is much smaller than those of the OMP algorithm (0.32 s) and IHT algorithm (0.17 s). Therefore, the data reconstruction speed of the proposed DCDA scheme is much faster than that of the data reconstruction algorithms in the CS theory. This is evidently because only simple matrix-vector multiplication and nonlinear activations are computed in the data reconstruction process of the DCDA scheme. On the contrary, cycles of complex iterations are required in the data reconstruction algorithms of the CS theory.

## 5 Conclusion

In this paper, we propose a DCDA scheme in WSNs, in which a DAE is trained with the historical sensed data, and its encoder and decoder parts are utilized to compress the sensed data and reconstruct the original data, respectively. In addition, a tree-based data collection algorithm was designed to collect the compressed data in a hybrid and cooperative mode. The experimental results show that our proposed method is superior to the existing CS-based data collection schemes and non-compression scheme in terms of data compression rate, average energy consumption, data reconstruction accuracy, and data reconstruction speed. In the future, we will focus on further improving the data compression and reconstruction performance of the proposed data collection scheme by introducing a time series prediction model.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61402094, 61572060, and 61702089), the Natural Science Foundation of Hebei Province (Nos. F2016501076 and F2016501079),

the Natural Science Foundation of Liaoning Province (No. 201602254), the Fundamental Research Funds for the Central Universities (No. N172304022), the Science and Technology Plan Project of Guangzhou (No. 201804010433), and the Bidding Project of Laboratory of Language Engineering and Computing (No. LEC2017ZBKT001).

## References

- [1] L. Borges, F. Velez, and A. Lebres, Survey on the characterization and classification of wireless sensor network applications, *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp.1860–1890, 2014.
- [2] F. Wang and J. Liu, Networked wireless sensor data collection: Issues, challenges, and approaches, *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 673–687, 2011.
- [3] L. Tang, J. Han, and G. Jiang, Mining sensor data in cyber-physical systems, *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 225–234, 2014.
- [4] G. Campobello, A. Segreto, and S. Serrano, Data gathering techniques for wireless sensor networks: A comparison, *International Journal of Distributed Sensor Networks*, vol. 2016, no. 1, pp. 1–17, 2016.
- [5] D. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [6] E. Candes and M. Wakin, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [7] R. Wang, J. Zhang, S. Ren, and Q. Li, A reducing iteration orthogonal matching pursuit algorithm for compressive sensing, *Tsinghua Science and Technology*, vol. 21, no. 1, pp. 71–79, 2016.
- [8] C. Luo, F. Wu, J. Sun, and C. Chen, Compressive data gathering for large-scale wireless sensor networks, in *Proc. 15th International Conference on Mobile Computing and Networking*, Beijing, China, 2009, pp. 145–156.
- [9] J. Luo, L. Xiang, and C. Rosenberg, Does compressed sensing improve the throughput of wireless sensor networks? in *Proc. 2010 IEEE International Conference on Communications*, Cape Town, South Africa, 2010, pp. 1–6.
- [10] J. Wang, S. Tang, B. Yin, and X. Li, Data gathering in wireless sensor networks through intelligent compressive sensing, in *Proc. 31st IEEE International Conference on Computer Communications*, Orlando, FL, USA, 2012, pp. 603–611.
- [11] J. Cheng, Q. Ye, H. Jiang, and D. Wang, STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks, *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 850–861, 2013.
- [12] L. Kong, M. Xia, X. Liu, G. Chen, Y. Gu, M. Wu, and X. Liu, Data loss and reconstruction in wireless sensor networks, *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 11, pp. 2818–2828, 2014.

- [13] L. Xiang, J. Luo, C. Deng, A. Vasilakos, and W. Lin, DECA: Recovering fields of physical quantities from incomplete sensory data, in *Proc. 9th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Seoul, Korea, 2012, pp. 182–190.
- [14] R. Middy, N. Chakravarty, and M. Naskar, Compressive sensing in wireless sensor networks—A survey, *IETE Technical Review*, vol. 33, no. 4, pp. 1–13, 2016.
- [15] D. Rumelhart, G. Hinton, and R. Williams, Learning representations by back-propagating errors, *Nature*, vol. 323, no. 9, pp. 533–536, 1986.
- [16] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [17] Y. Bengio, Learning deep architectures for AI, *Foundations & Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [18] G. Hinton and R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [19] P. Vincent, H. Larochelle, H. Bengio, and P. Manzagol, Extracting and composing robust features with denoising autoencoders, in *Proc. 25th International Conference on Machine Learning*, Helsinki, Finland, 2008, pp. 1096–1103.
- [20] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, in *Proc. 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011, pp. 833–840.
- [21] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, Variational autoencoder for deep learning of images, labels and captions, in *Proc. 30th Annual Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 2352–2360.
- [22] M. Li, T. Zhang, Y. Chen, and A. Smola, Efficient mini-batch training for stochastic optimization, in *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2014, pp. 661–670.
- [23] G. Li, J. He, and Y. Fu, Group-based intrusion detection system in wireless sensor networks, *Computer Communications*, vol. 31, no. 18, pp. 4324–4332, 2008.
- [24] J. Tropp and A. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [25] Intel Lab Data, <http://db.csail.mit.edu/labdata/labdata.html>, 2017.
- [26] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communication*, vol. 1, no. 4, pp. 660–670, 2002.
- [27] T. Blumensath and M. Davies, Iterative hard thresholding for compressed sensing, *Applied & Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2008.



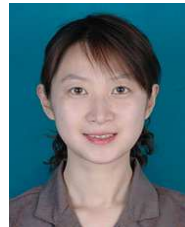
**Guorui Li** is currently an associate professor with School of Computer Science and Engineering, Northeastern University. He got the BS degree in 2001 from Hebei Normal University, ME degree in 2004 from Yanshan University, and PhD degree in 2009 from Beijing University of Technology. He visited

Virginia Polytechnic Institute and State University as a visiting scholar in 2015. His research interests include machine learning, compressed sensing, and internet of things.



**Sancheng Peng** is a full professor with School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China. He received the PhD degree in 2010 from Central South University, China. His research interests include network and information security, social networks,

trusted computing, and mobile computing.



**Ying Yuan** received the BE degree in 2004 from Northeastern University, MS degree in 2009 from Yanshan University, and PhD degree in 2015 from Northeastern University. She is currently a lecturer with School of Computer Science and Engineering, Northeastern University. Her main research interests include virtual

network embedding and virtual resource allocation in cloud computing.



**Cong Wang** received the BE degree in 2004 and MS degree in 2008 from Yanshan University, and PhD degree in 2011 from Northeastern University. He is currently an associate professor with School of Computer Science and Engineering, Northeastern University. His main research interests include virtual

network embedding, resource allocation in cloud computing, etc.



**Jianwei Niu** received the BS degree in 1991 from Zhengzhou Institute of Aeronautical Industry Management, MS degree in 1998 and PhD degree in 2002 from Beijing University of Aeronautics and Astronautics (BUAA, now Beihang University). He was a visiting scholar at School of Computer Science, Carnegie Mellon University, USA. He is a professor in the School of Computer Science and Engineering, BUAA. He has published more than 100 referred papers and filed more than 30 patents.

He has got five grants from NSFC, seven grants from National 863 Plan of China. He received the New Century Excellent Researcher Award from Ministry of Education of China in 2009, the first prize of technical invention of the Ministry of Education of China in 2012, Innovation Award from Nokia Research Center, and won the best paper award in IEEE ChinaCom 2014, ICC 2013, WCNC 2013, ICACT 2013, CWSN 2012, and GreenCom 2010. His current research interests include mobile and pervasive computing and mobile video analysis.